



ID	Member Name
S11199815	Savindya Fernando
S11198967	Karan Parmar

**CS341**  
**SOFTWARE QUALITY**  
**ASSURANCE AND**  
**TESTING**  
**SEMESTER 2, 2023**  
**ASSIGNMENT 2**

## TABLE OF CONTENTS

<b>ABSTRACT.....</b>	<b>2</b>
<b>INTRODUCTION.....</b>	<b>3</b>
<b>PART 1.....</b>	<b>4</b>
TASK 1 .....	4
TASK 2 .....	22
<b>PART 2.....</b>	<b>48</b>
TOOLS .....	48
<b>BIBLIOGRAPHY .....</b>	<b>54</b>
<b>MARK ALLOCATION SHEET .....</b>	<b>55</b>

## ABSTRACT

This report will discuss the Amazing Lunch Indicator Software Requirement Specification by using function point analysis and test design techniques. In software development, accurate metrics and effective testing methods are essential for project success. The first part of this document thoroughly analyzes the SRS, with a focus on section 3.1.1: User Interfaces. After carefully examining this section, it has been identified valuable insights into the input, output, queries, and file requirements of the user interfaces. Through calculating the Unadjusted Function Point Count and Adjusted Function Point metrics, we obtain a measurable indication of the size of the software. This metric is vital for understanding development efforts, estimating costs, and evaluating productivity levels.

In the second part, exploration of various techniques for designing test cases will be carried out. The UIs described in the SRS to create robust test cases using appropriate methodologies will be used. Decision tables are employed, if applicable, to identify input and output conditions and generate a comprehensive set of rules for each UI page and utilize methods to streamline these rules and optimize testing efficiency and effectiveness.

This integrated approach not only allows for quantifying software size but also establishes a thorough framework for testing. The ultimate goal is to enhance the software development process by providing valuable insights into size metrics and implementing a systematic testing methodology. Applying these crucial techniques within the real-world context of the Amazing Lunch Indicator project underscores their importance throughout all stages of software development.

## INTRODUCTION

The relationship between software development and testing is closely intertwined, as the quality of a software product greatly depends on effective testing methods. This document explores the important elements of function point analysis and test design techniques for the Amazing Lunch Indicator SRS. Function point analysis provides a quantitative measure that reveals the size of the software product, which contributes to resource allocation, cost estimation, and project management.

The initial portion of this task delves into the complexities of function point analysis, examining user interfaces to derive metrics such as Unadjusted Function Point Count and Adjusted Function Points. This quantitative evaluation establishes a basis for comprehending the magnitude and intricacy of the software product.

The second part shifts the focus to test design techniques, a critical component of ensuring software quality. The aim is to use appropriate methodologies and, if applicable, decision tables, to create comprehensive test cases for the user interfaces outlined in the SRS. The goal is to develop a robust testing framework that ensures the software meets its specified requirements and functions as intended.

Together, these two parts offer a holistic view of the interplay between software development, sizing metrics, and testing methodologies in the context of the Amazing Lunch Indicator project.

## PART 1

### TASK 1

#### DETAILED ANALYSIS OF FUNCTION POINT

#### WEIGHT FACTOR

Function Units	Low	Avg	High
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	5	7	10

## LOGIN PAGE

Software system components	Complexity Level		
	Simple	Average	Complex
User Input	Username: A single field for the user's username.	Username: A single field for the user's username with basic validation.	-
	Password: A single field for the user's password.	Password: A single field for the user's password with basic validation.	-
User Output	After entering the correct log-in credentials, the user is redirected to the search page.	After entering the correct log-in credentials, the user is redirected to the search page.	After entering the correct log-in credentials, the user is redirected to the search page.
User online Queries	Basic database query to check if the entered username and password match a registered user.	Additional query to check for the validity of the username and password (e.g., length, format).	Advanced queries to check for security measures, such as encryption and secure storage of log-in information.
Logical files	A basic log-in data file storing minimal user details.	An extended log-in data file with additional user information.	An encrypted log-in data file with advanced security measures.
External interface	-	-	-

## UNADJUSTED FUNCTION POINT COUNT

Software system components	Complexity Level									Total
	Simple			Average			Complex			
	Count	Weight Factor	Points	Count	Weight Factor	Points	Count	Weight Factor	Points	
User Inputs (UI)	2	× 3	6	2	× 4	8	0	× 6	0	14
Outputs (EO)	1	× 4	4	1	× 5	5	1	× 7	7	16
Online Inquiries (EQ)	1	× 3	3	1	× 4	4	1	× 6	6	13
Logical Files (ILF)	1	× 7	7	1	× 10	10	1	× 15	15	32
External Interfaces (EIF)	0	× 5	0	0	× 7	0	0	× 10	0	0
UFP										75

## SEARCH PAGE

Software system components	Complexity Level		
	Simple	Average	Complex
User Input	Free-text Search: A single field for free-text	Free-text Search: A single field for free-text with basic validation.	-
	Destination: Two fields for destination ( Min and Max)	Destination: Two fields for destination ( Min and Max) with basic validation.	-
	Price: Two fields for Price ( Min and Max)	Price: Two fields for Price ( Min and Max) with basic validation.	-
	Restaurant Type: A single field for type	Restaurant Type: A single field for type with basic validation.	-
	Specific Dish: A single field for Specific Dish	Specific Dish: A single field for Specific Dish with basic validation.	-
User Output	The user is redirected to the list view page.	The user is redirected to the list view page.	The user is redirected to the list view page.
User online Queries	-	-	Advanced queries to perform search based on the keyword and options
Logical files	-	-	An encrypted search results data file with advanced security measures.
External interface	-	-	-



## UNADJUSTED FUNCTION POINT COUNT

Software system components	Complexity Level									Total
	Simple			Average			Complex			
	Count	Weight Factor	Points	Count	Weight Factor	Points	Count	Weight Factor	Points	
User Inputs (UI)	7	× 3	21	7	× 4	28	0	× 6	0	49
Outputs (EO)	1	× 4	4	1	× 5	5	1	× 7	7	16
Online Inquiries (EQ)	0	× 3	0	0	× 4	0	1	× 6	6	6
Logical Files (ILF)	0	× 7	0	0	× 10	0	1	× 15	15	15
External Interfaces (EIF)	0	× 5	0	0	× 7	0	0	× 10	0	0
UFP										86

## PROFILE PAGE

Software system components	Complexity Level		
	Simple	Average	Complex
User Input	e-mail: A single field for the user's e-mail.	e-mail: A single field for the user's e-mail with basic validation.	e-mail: A single field for the user's e-mail with advanced validation (e.g., length, format).
	Password: A single field for the user's password.	Password: A single field for the user's password with basic validation.	Password: A single field for the user's password with advanced validation (e.g., strong password requirements).
	Phone Number: A single field for the user's Phone Number.	Phone Number: A single field for the user's Phone Number with basic validation.	Phone Number: A single field for the user's Phone Number with advanced validation (e.g., length, format).
	Select Language	Select Language with basic validation.	-
User Output	Changed user information	Changed user information	Changed user information
User online Queries	Basic database query to change the entered information	Additional query to check for the validity of the changed information	Advanced queries to check for security measures, such as encryption and secure storage of the changed information.
Logical files	A basic log-in data file storing minimal user details.	An extended log-in data file with additional user information.	An encrypted log-in data file with advanced security measures.

External interface	-	-	-
--------------------	---	---	---

## UNADJUSTED FUNCTION POINT COUNT

Software system components	Complexity Level									Total
	Simple			Average			Complex			
	Count	Weight Factor	Points	Count	Weight Factor	Points	Count	Weight Factor	Points	
User Inputs (UI)	4	× 3	12	4	× 4	16	3	× 6	18	46
Outputs (EO)	1	× 4	4	1	× 5	5	1	× 7	7	16
Online Inquiries (EQ)	1	× 3	3	1	× 4	4	1	× 6	6	13
Logical Files (ILF)	1	× 7	7	1	× 10	10	1	× 15	15	32
External Interfaces (EIF)	0	× 5	0	0	× 7	0	0	× 10	0	0
UFP										107

## LIST VIEW PAGE

Software system components	Complexity Level		
	Simple	Average	Complex
User Input	Price: Clickable button.	-	-
	Distance: Clickable button.	-	-
	Restaurant Type: Clickable button.	-	-
	Restaurant Name: Clickable button.	-	-
	Specific Dish: Clickable button.	-	-
User Output	List of search Results	List of search Results	List of search Results
User online Queries	-	-	Advanced query to perform search based on the keyword and options
Logical files	-	-	An encrypted search results data file with advanced security measures.
External interface	-	-	-

## UNADJUSTED FUNCTION POINT COUNT

Software system components	Complexity Level									Total
	Simple			Average			Complex			
	Count	Weight Factor	Points	Count	Weight Factor	Points	Count	Weight Factor	Points	
User Inputs (UI)	5	× 3	15	0	× 4	0	0	× 6	0	15
Outputs (EO)	1	× 4	4	1	× 5	5	1	× 7	7	16
Online Inquiries (EQ)	0	× 3	0	0	× 4	0	1	× 6	6	6
Logical Files (ILF)	0	× 7	0	0	× 10	0	1	× 15	15	15
External Interfaces (EIF)	0	× 5	0	0	× 7	0	0	× 10	0	0
UFP										52

## MAP VIEW PAGE

Software system components	Complexity Level		
	Simple	Average	Complex
User Input	Pin: Clickable Pin.	-	-
	Information Link: link.	-	-
User Output	Map of restaurants based on the search	Map of restaurants based on the search	Map of restaurants based on the search
	Information Link should show restaurant details	Information Link should show restaurant details	Information Link should show restaurant details
	Pin should redirect to the GPS.	Pin should redirect to the GPS.	Pin should redirect to the GPS.
User online Queries	-	-	Advanced query to perform search based on the keyword and options
Logical files	-	-	An encrypted search results data file with advanced security measures.
External interface	One external interfaces GPS	One external interfaces GPS	One external interfaces GPS

## UNADJUSTED FUNCTION POINT COUNT

Software system components	Complexity Level									Total
	Simple			Average			Complex			
	Count	Weight Factor	Points	Count	Weight Factor	Points	Count	Weight Factor	Points	
User Inputs (UI)	2	× 3	6	0	× 4	0	0	× 6	0	6
Outputs (EO)	3	× 4	12	3	× 5	15	3	× 7	21	48
Online Inquiries (EQ)	0	× 3	0	0	× 4	0	1	× 6	6	6
Logical Files (ILF)	0	× 7	0	0	× 10	0	1	× 15	15	15
External Interfaces (EIF)	1	× 5	5	1	× 7	7	1	× 10	10	22
UFP										97

## FILTER MENU PAGE

Software system components	Complexity Level		
	Simple	Average	Complex
User Input	Destination: One field for destination	Destination: One field for destination with basic validation.	-
	Price: One field for Price	Price: One field for Price with basic validation.	-
	Restaurant Type: A single field for type	Restaurant Type: A single field for type with basic validation.	-
	Specific Dish: A single field for Specific Dish	Specific Dish: A single field for Specific Dish with basic validation.	-
User Output	The filtered search results to be displayed.	The filtered search results to be displayed.	The filtered search results to be displayed.
User online Queries	-	-	Advanced queries to perform search based on the options
Logical files	-	-	An encrypted search results data file with advanced security measures.
External interface	-	-	-



## UNADJUSTED FUNCTION POINT COUNT

Software system components	Complexity Level									Total
	Simple			Average			Complex			
	Count	Weight Factor	Points	Count	Weight Factor	Points	Count	Weight Factor	Points	
User Inputs (UI)	4	× 3	12	4	× 4	16	0	× 6	0	28
Outputs (EO)	1	× 4	4	1	× 5	5	1	× 7	7	16
Online Inquiries (EQ)	0	× 3	0	0	× 4	0	1	× 6	6	6
Logical Files (ILF)	0	× 7	0	0	× 10	0	1	× 15	15	15
External Interfaces (EIF)	0	× 5	0	0	× 7	0	0	× 10	0	0
UFP										65

## WEB PORTAL PAGE

Software system components	Complexity Level		
	Simple	Average	Complex
User Input	Username: A single field for the user's username.	Username: A single field for the user's username with basic validation.	Username: A single field for the user's username with advanced validation (e.g., length, format).
	Password: A single field for the user's password.	Password: A single field for the user's password with basic validation.	Password: A single field for the user's password with advanced validation (e.g., strong password requirements).
	User type check Button	User type check Button	User type check Button
User Output	After entering the correct log-in credentials, the user can edit restaurant or user information.	After entering the correct log-in credentials, the user can edit restaurant or user information.	After entering the correct log-in credentials, the user can edit restaurant or user information.
User online Queries	Basic database query to check if the entered username and password match a registered user.	Additional query to check for the validity of the username and password (e.g., length, format).	Advanced queries to check for security measures, such as encryption and secure storage of log-in information.
Logical files	A basic log-in data file storing minimal user details.	An extended log-in data file with additional user information.	An encrypted log-in data file with advanced security measures.
External interface	-	-	-

## UNADJUSTED FUNCTION POINT COUNT

Software system components	Complexity Level									Total
	Simple			Average			Complex			
	Count	Weight Factor	Points	Count	Weight Factor	Points	Count	Weight Factor	Points	
User Inputs (UI)	3	× 3	9	3	× 4	12	3	× 6	18	39
Outputs (EO)	1	× 4	4	1	× 5	5	1	× 7	7	16
Online Inquiries (EQ)	1	× 3	3	1	× 4	4	1	× 6	6	13
Logical Files (ILF)	1	× 7	7	1	× 10	10	1	× 15	15	32
External Interfaces (EIF)	0	× 5	0	0	× 7	0	0	× 10	0	0
UFP										100

## OVERALL FUNCTION POINT ANALYSIS

### OVERALL UNADJUSTED FUNCTION POINT COUNT

UI	UFP
Login	75
Search	86
Profile	107
List View	52
Map View	97
Filter Menu	65
Web Portal	100
<b>Total</b>	<b>582</b>

### ADJUSTED FUNCTION POINT COUNT

The relative complexity adjustment factor is calculated based on 14 General System Characteristics that may influence the complexity of a software project. Each GSC is rated on a scale from 0 to 5, with higher ratings indicating a greater degree of influence (Function Point Analysis - Introduction and Fundamentals, 2020).

### VALUE ADJUSTMENT FACTORS

0	No Influences
1	Incidental
2	Moderate
3	Average
4	Significant
5	Essential

## VALUE ADJUSTMENT FORM

Number	Subject	Grade
1	Data communications	4
2	Distributed data processing	3
3	Performance	3
4	Heavily used configuration	2
5	Transaction rate	4
6	Online data entry	3
7	End-user efficiency	4
8	Online update	4
9	Complex processing	3
10	Reusability	2
11	Installation ease	2
12	Operational ease	3
13	Multiple sites	1
14	Facilitate change	3
Total		41

## OVERALL ADJUSTED FUNCTION POINT COUNT

$$\begin{aligned}\mathbf{AFP} &= \text{UFP} \times (0.65 + 0.01 \times \text{RCAF}) \\ &= 582 \times (0.65 + 0.01 \times 41) \\ &= 582 \times 1.06 \\ &= \mathbf{616.92}\end{aligned}$$

## FUNCTION POINT ANALYSIS IN ESTIMATING THE EFFORT, COST AND PRODUCTIVITY OF THE SOFTWARE DEVELOPMENT

Function Point Analysis (FPA), which offers a systematic and quantitative technique of assessing effort, cost, and productivity, is an essential methodology in software development. Function Points (FP) are a statistic produced by Function Point Analyses (FPA), which measure a software system's functionality by counting inputs, outputs, queries, files, and interfaces. This measure provides the foundation for assessing development effort. Each functional component's complexity is assessed, and subsequent effort estimating models transform Function Points into person-months or person-hours, establishing the basis for resource planning. Cost estimation is therefore made easier by correlating monetary values with projected work, taking into account elements like labor rates and project-specific charges.

FPA not only acts as a tool for financial planning, but it also adds to productivity assessment by assessing the development team's efficiency. The ratio of Function Points to effort invested reveals how efficiently functionality is given. Furthermore, FPA promotes benchmarking against previous projects and industry norms, assists in project control by creating baselines for progress tracking, and improves communication among stakeholders during negotiations. In summary, Function Point Analysis is a priceless tool that enables software development teams and organizations to make educated decisions, optimize resource allocation, and constantly improve their development processes.

## TASK 2

The test technique used for this task is **Boundary value Analysis** as the question involves input ranges and constraints. The UI's requires users to input values within certain limits, test at the boundaries of these limits to ensure the system handles them correctly.

### LOGIN PAGE TEST CASES

Test Case #	Test case name	Pre-Conditions	Input	Expected Result	Reference to Detailed Design / Spec Document
1	Enter Valid Username and Valid Password	Registered User	Valid Username Valid Password	Login Successful	3.1.1 Figure 2
2	Enter Valid Username and Invalid Password	Registered User	Valid Username Invalid Password	Login Unsuccessful	3.1.1 Figure 2
3	Enter Invalid Username and valid Password	Registered User	Invalid Username Valid Password	Login Unsuccessful	3.1.1 Figure 2
4	Enter Invalid Username and Invalid Password	Registered User	Invalid Username Invalid Password	Login Unsuccessful	3.1.1 Figure 2
5	Enter Valid Username and Valid Password	A First-time User	Valid Username Valid Password	Login Unsuccessful (Should Register)	3.1.1 Figure 2
6	Enter Invalid Username and Valid Password	A First-time User	Invalid Username Valid Password	Login Unsuccessful (Should Register)	3.1.1 Figure 2
7	Enter Valid Username and Invalid Password	A First-time User	Valid Username Invalid Password	Login Unsuccessful (Should Register)	3.1.1 Figure 2
8	Enter Invalid Username and Invalid Password	A First-time User	Invalid Username Invalid Password	Login Unsuccessful (Should Register)	3.1.1 Figure 2

## DECISION TABLE

Condition	Inputs							
	1	2	3	4	5	6	7	8
Valid Username	T	T	T	T	F	F	F	F
Valid Password	T	T	F	F	T	T	F	F
Registered User	T	F	T	F	T	F	T	F
Output Result	Login Successful	Login Unsuccessful	Login Unsuccessful	Login Unsuccessful	Login Unsuccessful	Login Unsuccessful	Login Unsuccessful	Login Unsuccessful

## DECISION TABLE AFTER RULE REDUCTION

Rule reduction can be applied to the above decision Table.

Condition : Regardless of the validity of the inputs entered by the user, if the user is not Registered, hence the login will be unsuccessful.

Condition	Inputs				
	1	2	3	4	5
Valid Username	T	T	F	F	T or F
Valid Password	T	F	T	F	T or F
Registered User	T	T	T	T	F
Output Result	Login Successful	Login Unsuccessful	Login Unsuccessful	Login Unsuccessful	Login Unsuccessful



## SEARCH PAGE TEST CASES

Test Case	Test case name	Pre-Conditions	Input	Expected Result	Reference to Spec Document
1	Enter Free-text to search	Registered User	Free-text	Search Successful	3.1.1 Figure 3
2	Leave all options empty	Registered User	Null	Search Unsuccessful	3.1.1 Figure 3
3	Select only Destination option with <b>valid inputs</b>	Registered User	Select destination Option Valid inputs	Search Successful	3.1.1 Figure 3
4	Select only Destination option with <b>invalid inputs</b>	Registered User	Select destination Option Invalid inputs	Search Unsuccessful	3.1.1 Figure 3
5	Select only Destination option with <b>invalid inputs</b> and <b>Free-text with valid inputs.</b>	Registered User	Select destination Option Invalid inputs Valid Free-Text	Search Successful	3.1.1 Figure 3
5	Select only Destination option with <b>valid inputs</b> and <b>Free-text with Invalid inputs.</b>	Registered User	Select destination Option Valid inputs Invalid Free-Text	Search Successful	3.1.1 Figure 3
6	Select only the Price option with <b>valid inputs.</b>	Registered User	Select price Option Valid inputs	Search Successful	3.1.1 Figure 3
7	Select only the Price option with <b>invalid inputs.</b>	Registered User	Select price Option Invalid inputs	Search Successful	3.1.1 Figure 3
8	Select only price option with <b>invalid inputs</b> and <b>Free-text with valid inputs.</b>	Registered User	Select price Option Invalid inputs Valid Free-Text	Search Successful	3.1.1 Figure 3

5	Select only price option with <b>valid inputs</b> and <b>Free-text with invalid inputs.</b>	Registered User	Select price Option valid inputs Invalid Free-Text	Search Successful	3.1.1 Figure 3
9	Select only Restaurant Type option with <b>valid inputs</b>	Registered User	Select Restaurant Option Valid inputs	Search Successful	3.1.1 Figure 3
10	Select only Restaurant Type option with <b>invalid inputs</b>	Registered User	Select Restaurant Type Option Valid inputs	Search Unsuccessful	3.1.1 Figure 3
11	Select only Restaurant Type option with <b>invalid inputs</b> and <b>Free-text with valid inputs.</b>	Registered User	Select Restaurant Type Option Invalid inputs Valid Free-Text	Search Successful	3.1.1 Figure 3
11	Select only Restaurant Type option with <b>valid inputs</b> and <b>Free-text with invalid inputs.</b>	Registered User	Select Restaurant Type Option Valid inputs Invalid Free-Text	Search Successful	3.1.1 Figure 3
12	Select only Dish option with <b>valid inputs</b>	Registered User	Select dish Option Valid inputs	Search Successful	3.1.1 Figure 3
13	Select only Dish option with <b>invalid inputs</b>	Registered User	Select dish Option Invalid inputs	Search Unsuccessful	3.1.1 Figure 3
14	Select only Dish option with <b>invalid inputs</b> and <b>Free-text with valid inputs.</b>	Registered User	Select dish Option Invalid inputs Valid Free-Text	Search Successful	3.1.1 Figure 3

15	Select any two options with <b>valid inputs</b>	Registered User	Select Options valid option input	Search Successful	3.1.1 Figure 3
14	Select only Dish option with <b>valid inputs</b> and <b>Free-text with invalid inputs</b> .	Registered User	Select dish Option Valid inputs Invalid Free-Text	Search Successful	3.1.1 Figure 3
16	Select any two options with <b>invalid inputs</b>	Registered User	Select Options invalid option input	Search Unsuccessful	3.1.1 Figure 3
17	Select any two options with <b>invalid inputs</b> and <b>Free-text with valid inputs</b>	Registered User	Select Options invalid option input Valid Free-Text	Search Successful	3.1.1 Figure 3
18	Select any three options with <b>valid inputs</b>	Registered User	Select Options valid option input	Search Successful	3.1.1 Figure 3
19	Select any three options with <b>invalid inputs</b>	Registered User	Select Options invalid option input	Search Unsuccessful	3.1.1 Figure 3
20	Select any three options with <b>invalid inputs</b> and <b>Free-text with valid inputs</b>	Registered User	Select Options invalid option input valid Free-Text	Search Unsuccessful	3.1.1 Figure 3
21	Select all options with <b>valid inputs</b>	Registered User	Select Options valid option input	Search Successful	3.1.1 Figure 3
22	Select all options with <b>invalid inputs</b>	Registered User	Select Options invalid option input	Search Unsuccessful	3.1.1 Figure 3
23	Select all options with <b>valid inputs</b> and <b>Free-text with valid inputs</b>	Registered User	Select Options valid option input valid Free-text	Search Successful	3.1.1 Figure 3
24	Select all options with <b>valid inputs</b> and <b>Free-text</b>	Registered User	Select Options valid option input Invalid Free-text	Search Successful	3.1.1 Figure 3

	<b>with invalid inputs</b>				
25	Select all options with <b>invalid inputs</b> and <b>Free-text with valid inputs</b>	Registered User	Select Options Invalid option input valid Free-text	Search Successful	3.1.1 Figure 3
26	Select all options with <b>invalid inputs</b> and <b>Free-text with invalid inputs</b>	Registered User	Select Options Invalid option input Invalid Free-text	Search Unsuccessful	3.1.1 Figure 3
27	No Search Found	Registered User	Select Options Enter Free-text	1. User should be informed but kept on the search page 2. Can conduct a new search	3.1.1 Figure 3

## DECISION TABLE

Test Case	Condition					Output
	Free-Text Search	Destination	Price	Type	Dish	
1	T	T	T	T	T	Search Successful
2	T	T	T	F	F	Search Unsuccessful
3	T	T	F	T	F	Search Successful
4	T	F	F	F	F	Search Unsuccessful
5	T	-	-	-	-	Search Successful
6	T	T	-	-	-	Search Successful
7	T	F	-	-	-	Search Successful
8	T	-	T	-	-	Search Successful
9	T	-	F	-	-	Search Successful
10	T	-	-	T	-	Search Successful
11	T	-	-	F	-	Search Successful
12	T	-	-	-	T	Search Unsuccessful
13	T	-	-	-	F	Search Successful
14	T	T	T	-	-	Search Successful
15	T	T	F	-	-	Search Successful
16	T	F	T	-	-	Search Unsuccessful
17	T	F	F	-	-	Search Successful
18	T	-	T	T	-	Search Successful
19	T	-	T	F	-	Search Successful

20	T	-	F	T	-	Search Unsuccessful
21	T	-	F	F	-	Search Successful
22	T	-	-	T	T	Search Successful
23	T	-	-	T	F	Search Unsuccessful
24	T	-	-	F	T	Search Unsuccessful
25	T	-	-	F	F	Search Successful
26	F	T	T	T	T	Search Unsuccessful
27	F	T	T	F	F	Search Successful
28	F	T	F	T	F	Search Successful
29	F	F	F	F	F	Search Successful
30	F	-	-	-	-	Search Unsuccessful
31	F	T	-	-	-	Search Successful
32	F	F	-	-	-	Search Unsuccessful
33	F	-	T	-	-	Search Successful
34	F	-	F	-	-	Search Unsuccessful
35	F	-	-	T	-	Search Successful
36	F	-	-	F	-	Search Successful
37	F	-	-	-	T	Search Successful
38	F	-	-	-	F	Search Successful
39	F	T	T	-	-	Search Successful
40	F	T	F	-	-	Search Successful
41	F	F	T	-	-	Search Successful
42	F	F	F	-	-	Search Unsuccessful
43	F	-	T	T	-	Search Successful
44	F	-	T	F	-	Search Successful
45	F	-	F	T	-	Search Successful
46	F	-	F	F	-	Search Unsuccessful
47	F	-	-	T	T	Search Successful
48	F	-	-	T	F	Search Successful
49	F	-	-	F	T	Search Successful
50	F	-	-	F	F	Search Unsuccessful
51	-	-	-	-	-	User should be informed but kept on the search page. Can conduct a new search

## DECISION TABLE AFTER RULE REDUCTION

Rule reduction can be applied to the above decision Table.

Condition : Regardless of the selected options in search and the option inputs entered by the user, if the Free-text search is valid, the search will be successful.

Test Case	Condition					Output
	Free-Text Search	Destination	Price	Type	Dish	
1	T	T/F/-	T/F/-	T/F/-	T/F/-	Search Successful
2	F	T	T	T	T	Search Unsuccessful
3	F	T	T	F	F	Search Successful
4	F	T	F	T	F	Search Successful
5	F	F	F	F	F	Search Successful
6	F	-	-	-	-	Search Unsuccessful
7	F	T	-	-	-	Search Successful
8	F	F	-	-	-	Search Unsuccessful
9	F	-	T	-	-	Search Successful
10	F	-	F	-	-	Search Unsuccessful
11	F	-	-	T	-	Search Successful
12	F	-	-	F	-	Search Successful
13	F	-	-	-	T	Search Successful
14	F	-	-	-	F	Search Successful
15	F	T	T	-	-	Search Successful
16	F	T	F	-	-	Search Successful
17	F	F	T	-	-	Search Successful
18	F	F	F	-	-	Search Unsuccessful
19	F	-	T	T	-	Search Successful
20	F	-	T	F	-	Search Successful
21	F	-	F	T	-	Search Successful
22	F	-	F	F	-	Search Unsuccessful
23	F	-	-	T	T	Search Successful
24	F	-	-	T	F	Search Successful
25	F	-	-	F	T	Search Successful
26	F	-	-	F	F	Search Unsuccessful
27	-	-	-	-	-	User should be informed but kept on the search page. Can conduct a new search

## PROFILE PAGE TEST CASES

Test Case	Test case name	Pre-Conditions	Input	Expected Result	Reference to Spec Document
1	Enter Valid E-mail to edit	Registered User	Valid E-mail	Can Edit	3.1.1 Figure 4
2	Enter Invalid E-mail to edit	Registered User	Invalid E-mail	Cannot Edit	3.1.1 Figure 4
3	Enter Valid Phone Number to edit	Registered User	Valid Phone Number	Can Edit	3.1.1 Figure 4
4	Enter Invalid Phone Number to edit	Registered User	Invalid Phone Number	Cannot Edit	3.1.1 Figure 4
5	Change Password with valid input	Registered User	Valid Password	Password Changed Successfully	3.1.1 Figure 4
6	Change Password with invalid input	Registered User	Invalid Password	Password Changed Unsuccessfully	3.1.1 Figure 4
7	Select Language to change	Registered User	Select Language	Language changed	3.1.1 Figure 4
8	Change language without selecting an option to change	Registered User	NULL	Language cannot be changed	3.1.1 Figure 4
9	Enter Valid E-mail to edit	First-time User	Valid E-mail	Can Edit (Should Register)	3.1.1 Figure 4
10	Enter Invalid E-mail to edit	First-time User	Invalid E-mail	Cannot Edit (Should Register)	3.1.1 Figure 4
11	Enter Valid Phone Number to edit	First-time User	Valid Phone Number	Can Edit (Should Register)	3.1.1 Figure 4
12	Enter Invalid Phone Number to edit	First-time User	Invalid Phone Number	Cannot Edit (Should Register)	3.1.1 Figure 4
13	Change Password with valid input	First-time User	Valid Password	Cannot change password (Should Register)	3.1.1 Figure 4
14	Change Password with invalid input	First-time User	Invalid Password	Cannot change password (Should Register)	3.1.1 Figure 4
15	Select Language to change	Not a First-time User	Select Language	Language changed (Should Register)	3.1.1 Figure 4
16	Change language without selecting an option to change	Not a First-time User	NULL	Language cannot be changed (Should Register)	3.1.1 Figure 4

## DECISION TABLE

Test Case	Condition					Expected Output
	Registered User	Valid Email	Valid Phone Number	Valid Password	Select Language	
1	T	T	-	-	-	E-mail Edited
2	T	F	-	-	-	E-mail cannot be Edited (Invalid E-mail)
3	T	-	T	-	-	Phone Number Edited
4	T	-	F	-	-	Phone Number cannot be Edited (Invalid Phone Number)
5	T	-	-	T	-	Change Password
6	T	-	-	F	-	Password cannot be Changed (Invalid Password)
7	T	-	-	-	T	Language Changed
8	T	-	-	-	F	Language cannot be changed
9	F	T	-	-	-	Phone Number cannot be Edited (Should Register)
10	F	F	-	-	-	Phone Number cannot be Edited (Should Register)
11	F	-	T	-	-	Phone Number cannot be Edited (Should Register)
12	F	-	F	-	-	Phone Number cannot be Edited (Should Register)
13	F	-	-	T	-	Password cannot be Changed (Should Register)
14	F	-	-	F	-	Password cannot be Changed (Should Register)
15	F	-	-	-	T	Language cannot be changed (Should Register)
16	F	-	-	-	F	Language cannot be changed (Should Register)



## DECISION TABLE AFTER RULE REDUCTION

Rule reduction can be applied to the above decision Table.

Condition : Regardless of the selected options in search and the option to edit, if the user is a first-time user they cannot edit.

Test Case	Condition					Expected Output
	Registered User	Valid Email	Valid Phone Number	Valid Password	Select Language	
1	T	T	-	-	-	E-mail Edited
2	T	F	-	-	-	E-mail cannot be Edited (Invalid E-mail)
3	T	-	T	-	-	Phone Number Edited
4	T	-	F	-	-	Phone Number cannot be Edited (Invalid Phone Number)
5	T	-	-	T	-	Change Password
6	T	-	-	F	-	Password cannot be Changed (Invalid Password)
7	T	-	-	-	T	Language Changed
8	T	-	-	-	F	Language cannot be changed
9	F	T/F/-	T/F/-	T/F/-	T/F/-	Cannot Edit (Should Register)

## LIST VIEW TEST CASES

Test Case	Test case name	Pre-Conditions	Input	Expected Result	Reference to Spec Document
1	Sorting by Price	Perform a search by price	Click price to sort	<ol style="list-style-type: none"> <li>1. Results should be sorted by average price, then distance, restaurant type, and specific dish.</li> <li>2. Results should be ordered from cheapest to most expensive.</li> </ol>	3.1.1 Figure 5
2	Sorting by Restaurant name	Perform a search by Restaurant name	Click Restaurant name to sort	<ol style="list-style-type: none"> <li>1. Results should be sorted by Restaurant name, then distance, average price, restaurant type, and specific dish.</li> <li>2. Results should be ordered alphabetically.</li> </ol>	3.1.1 Figure 5
3	Sorting by restaurant type	Perform a search by restaurant type	Click restaurant type to sort	<ol style="list-style-type: none"> <li>1. Results should be sorted by Restaurant type, then distance, average price, restaurant name, and specific dish.</li> <li>2. Results should be ordered alphabetically.</li> </ol>	3.1.1 Figure 5
4	Sorting by specific dish	Perform a search by specific dish	Click specific dish to sort	<ol style="list-style-type: none"> <li>1. Results should be sorted by specific dish, then distance, average price, restaurant type, and restaurant name.</li> <li>2. Results should be ordered alphabetically.</li> </ol>	3.1.1 Figure 5
5	Sorting by distance	Perform a search by distance	Click distance to sort	<ol style="list-style-type: none"> <li>1. Results should be sorted by distance, then average price, restaurant type, and specific dish.</li> <li>2. Results should be ordered from closets to furthest distance according to the user's position.</li> </ol>	3.1.1 Figure 5
6	Sorting by Price again	Perform a search by price	Click price again to sort	<ol style="list-style-type: none"> <li>1. Results should be sorted by average price, then distance, restaurant type, and specific dish.</li> </ol>	3.1.1 Figure 5

				2. Results should be ordered from cheapest to most expensive.	
7	Sorting by Restaurant name again	Perform a search by Restaurant name	Click Restaurant name again to sort	<ol style="list-style-type: none"> <li>1. Results should be sorted by Restaurant name, then distance, average price, restaurant type, and specific dish.</li> <li>2. Results should be in reversed order alphabetically.</li> </ol>	3.1.1 Figure 5
8	Sorting by restaurant type again	Perform a search by restaurant type	Click restaurant type again to sort	<ol style="list-style-type: none"> <li>1. Results should be sorted by Restaurant type, then distance, average price, restaurant name, and specific dish.</li> <li>2. Results should be in reversed order alphabetically.</li> </ol>	3.1.1 Figure 5
9	Sorting by specific dish again	Perform a search by specific dish	Click specific dish again to sort	<ol style="list-style-type: none"> <li>1. Results should be sorted by specific dish, then distance, average price, restaurant type, and restaurant name.</li> <li>2. Results should be in reversed order alphabetically.</li> </ol>	3.1.1 Figure 5
10	Sorting by distance again	Perform a search by distance	Click distance to sort	<ol style="list-style-type: none"> <li>1. Results should be sorted by distance, then average price, restaurant type, and specific dish.</li> <li>2. Results should be ordered from furthest to closets distance according to the user's position.</li> </ol>	3.1.1 Figure 5
11	Maximum Display Limit	Search with an option that yields more than 100 results.	Click an option with 100 results to sort	Display only the first 100 results, allow scrolling for more.	3.1.1 Figure 5
12	Toggle between map and list view.	Perform a search and current page is the list view	Click map view	Switch between the map and list view work well for all search options.	3.1.1 Figure 5

13	Toggle between map and list view.	Current page is the map view	Click list view	Switch between the map and list view work well for all search options.	3.1.1 Figure 5
----	-----------------------------------	------------------------------	-----------------	--	-------------------

## DECISION TABLE

Condition		Options						Expected output
		Price	Restaurant Name	Restaurant type	Distance	Specific dish	Results >100	
1	Sorting by Price	T	-	-	-	-	-	Results should be sorted by average price, then distance, restaurant type, and specific dish.  Results should be ordered from cheapest to most expensive.
2	Click Sorting by Price again	T	-	-	-	-	-	Results should be sorted by average price, then distance, restaurant type, and specific dish.  Results should be in reverse order from expensive to the cheapest.

3	Sorting by Restaurant name	-	T	-	-	-	-	<p>Results should be sorted by Restaurant name, then distance, average price, restaurant type, and specific dish.</p> <p>Results should be in ascending order alphabetically.</p>
4	Click Sorting by Restaurant name again	-	T	-	-	-	-	<p>Results should be sorted by Restaurant name, then distance, average price, restaurant type, and specific dish.</p> <p>Results should be in descending order alphabetically.</p>
5	Sorting by restaurant type	-	-	T	-	-	-	<p>Results should be sorted by Restaurant type, then distance, average price, restaurant name, and specific dish.</p> <p>Results should be in ascending order alphabetically.</p>
6	Click Sorting by restaurant type again	-	-	T	-	-	-	<p>Results should be sorted by Restaurant type, then distance, average price, restaurant name, and specific dish.</p>

								Results should be in descending order alphabetically.
7	Sorting by distance	-	-	-	T	-	-	Results should be sorted by distance, then average price, restaurant type, and specific dish.  Results should be ordered from closets to furthest distance according to the user's position.
8	Click Sorting by distance again	-	-	-	T	-	-	Results should be sorted by distance, then average price, restaurant type, and specific dish.  Results should be ordered from furthest to closets distance according to the user's position.
9	Sorting by specific dish	-	-	-	-	T	-	Results should be sorted by specific dish, then distance, average price, restaurant type, and restaurant name.  Results should be in ascending order alphabetically.
10	Click Sorting by	-	-	-	-	T	-	Results should be sorted by specific dish, then distance, average price,

	specific dish again							restaurant type, and restaurant name.  Results should be in descending order alphabetically.
11	Search results > 100	-	-	-	-	-	T	Display only the first 100 results, allow scrolling for more.

## MAP VIEW TEST CASES

Test Case	Test case name	Pre-Conditions	Input	Expected Result	Reference to Spec Document
1	View search results in map	User should click Map view	Search option	Displays the relevant and closest restaurants according to the user's position.	3.1.1 Figure 6
2	Click the information link	User should click Map view	Click the relevant information link	direct the user to an information page	3.1.1 Figure 6
3	Select a pin	User should click Map view	Click a pin	Location of the restaurant should be sent to the mobile phone's GPS	3.1.1 Figure 6
4	Reached the Destination	User should click Map view	-	A user should be able to go back to the search page on the mobile application	3.1.1 Figure 6
5	Maximum Display Limit > 100	Search with an option that yields more than 100 results.	Click an option with 100 results to sort	Display only 100 results.	3.1.1 Figure 6
6	Toggle between map and list view.	Perform a search and current page is the list view	Click map view	Switch between the map and list view work well for all search options.	3.1.1 Figure 6
7	Toggle between map and list view again	Current page is the map view	Click list view	Switch between the map and list view work well for all search options.	3.1.1 Figure 6

## FILTER MENU TEST CASES

Test Case	Test case name	Pre-Conditions	Input	Expected Result	Reference to Spec Document
1	Increase or decrease price, distance and Select restaurant type and specific dish with valid inputs	User should click filter menu	Valid integer for price and distance.  Valid option from restaurant type and specific dish	Results should be filtered up based on the ranges and types.	3.1.1 Figure 7
2	Increase or decrease price, distance and Select restaurant type and specific dish with invalid inputs	User should click filter menu	Invalid integer for price and distance.  Invalid option from restaurant type and specific dish	Error	3.1.1 Figure 7
3	Increase or decrease price, distance with valid inputs and Select restaurant type and specific dish with invalid inputs	User should click filter menu	Valid integer for price and distance.  Invalid option from restaurant type and specific dish	Error	3.1.1 Figure 7
4	Increase or decrease price, distance with invalid inputs and Select restaurant type and specific dish with valid inputs	User should click filter menu	Invalid integer for price and distance.  Valid option from restaurant type and specific dish	Error	3.1.1 Figure 7
5	Increase or decrease price with invalid input, distance with valid input and Select restaurant type and specific dish with valid inputs	User should click filter menu	Invalid integer for price  Valid integer for distance  Valid option from restaurant type and specific dish	Error	3.1.1 Figure 7
6	Increase or decrease price with valid input,	User should click filter menu	Valid integer for price	Error	3.1.1 Figure 7



	distance with invalid input and Select restaurant type and specific dish with valid inputs		Invalid integer for distance  Valid option from restaurant type and specific dish		
7	Increase or decrease price, distance with valid inputs Select a restaurant type to filter with valid inputs and specific dish with invalid inputs	User should click filter menu	Valid integer for price, distance  Valid option from restaurant type  Invalid input for specific dish	Error	3.1.1 Figure 7
8	Increase or decrease price, distance with valid inputs Select a restaurant type to filter with invalid inputs and specific dish with valid inputs	User should click filter menu	Valid integer for price, distance  Invalid option from restaurant type  Valid input for specific dish	Error	3.1.1 Figure 7
9	Increase or decrease price, distance with invalid inputs Select a restaurant type to filter with valid inputs and specific dish with invalid inputs	User should click filter menu	Invalid integer for price, distance  Valid option from restaurant type  Invalid input for specific dish	Error	3.1.1 Figure 7
10	Increase or decrease price, distance with invalid inputs Select a restaurant type to filter with invalid inputs and specific dish with valid inputs	User should click filter menu	Invalid integer for price, distance  Invalid option from restaurant type  Valid input for specific dish	Error	3.1.1 Figure 7
11	Increase or decrease price with invalid input, distance with valid	User should click filter menu	Invalid integer for price  Valid integer for distance	Error	3.1.1 Figure 7

	input and Select restaurant type with valid input and specific dish with invalid input		Valid option from restaurant type  Invalid option from specific dish		
12	Increase or decrease price with invalid input, distance with valid input and Select restaurant type with invalid input and specific dish with valid input	User should click filter menu	Invalid integer for price  Valid integer for distance  Invalid option from restaurant type  Valid option from specific dish	Error	3.1.1 Figure 7
13	Increase or decrease price with invalid input, distance with valid input and Select restaurant type and specific dish with invalid inputs	User should click filter menu	Invalid integer for price  Valid integer for distance  Invalid option from restaurant type  Invalid option from specific dish		3.1.1 Figure 7
14	Increase or decrease price with valid input, distance with invalid input and Select restaurant type with valid input and specific dish with invalid input	User should click filter menu	Valid integer for price  Invalid integer for distance  Valid option from restaurant type  Invalid option from specific dish	Error	3.1.1 Figure 7
15	Increase or decrease price with valid input, distance with invalid input and Select restaurant type with invalid input and specific dish with valid input	User should click filter menu	Valid integer for price  Invalid integer for distance  Invalid option from restaurant type  Valid option from specific dish	Error	3.1.1 Figure 7

16	Increase or decrease price with valid input, distance with invalid input and Select restaurant type specific dish with invalid input	User should click filter menu	Valid integer for price  Invalid integer for distance  Invalid option from restaurant type  Invalid option from specific dish	Error	3.1.1 Figure 7
----	--	-------------------------------	---	-------	-------------------

## DECISION TABLE

Test Case	Condition				Expected Output
	Valid price range	Valid distance	Valid restaurant type	Valid specific dish	
1	T	T	T	T	Results should be filtered up based on the ranges and types.
2	T	T	T	F	Error
3	T	T	F	T	Error
4	T	T	F	F	Error
5	T	F	T	T	Error
6	T	F	T	F	Error
7	T	F	F	T	Error
8	T	F	F	F	Error
	F	T	T	T	Error
10	F	T	T	F	Error
11	F	T	F	T	Error
12	F	T	F	F	Error
13	F	F	T	T	Error

14	F	F	T	F	Error
15	F	F	F	T	Error
16	F	F	F	F	Error

## WEB PORTAL TEST CASES

Test Case #	Test case name	Pre-Conditions	Input	Expected Result	Reference to Spec Document
1	Enter Valid restaurant owner Username and Valid Password  Select the restaurant owner choice	Registered restaurant owner	Valid Username Valid Password Valid choice	Login successful as a restaurant owner	3.1.1 Figure 8
2	Enter Valid restaurant owner Username and Invalid Password  Select the restaurant owner choice	Registered restaurant owner	Valid Username Invalid Password Valid choice	Login unsuccessful as a restaurant owner	3.1.1 Figure 8
3	Enter Invalid restaurant owner Username and valid Password  Select the restaurant owner choice	Registered restaurant owner	Invalid Username Valid Password Valid choice	Login unsuccessful as a restaurant owner	3.1.1 Figure 8
4	Enter Invalid restaurant owner Username and Invalid Password  Select the restaurant owner choice	Registered restaurant owner	Invalid Username Invalid Password Valid choice	Login unsuccessful as a restaurant owner	3.1.1 Figure 8
5	Enter Valid restaurant owner Username and Valid Password  Select the admin choice	Registered restaurant owner	Valid Username Valid Password Invalid choice	Login unsuccessful as an admin	3.1.1 Figure 8

6	Enter Valid restaurant owner Username and Invalid Password  Select the admin choice	Registered restaurant owner	Valid Username Invalid Password Invalid choice	Login unsuccessful as an admin	3.1.1 Figure 8
7	Enter Invalid restaurant owner Username and valid Password  Select the admin choice	Registered restaurant owner	Invalid Username Valid Password Invalid choice	Login unsuccessful as an admin	3.1.1 Figure 8
8	Enter Invalid restaurant owner Username and Invalid Password  Select the admin choice	Registered restaurant owner	Invalid Username Invalid Password Invalid choice	Login unsuccessful as an admin	3.1.1 Figure 8
9	Enter valid admin Username and valid Password  Select the admin choice	Registered admin	Valid Username Valid Password Valid choice	Login successful as an admin	3.1.1 Figure 8
10	Enter Invalid admin Username and valid Password  Select the admin choice	Registered admin	Invalid Username Valid Password Valid choice	Login unsuccessful as an admin	3.1.1 Figure 8
11	Enter valid admin Username and invalid Password  Select the admin choice	Registered admin	Valid Username Invalid Password Valid choice	Login unsuccessful as an admin	3.1.1 Figure 8
12	Enter invalid admin Username and invalid Password  Select the admin choice	Registered admin	Invalid Username Invalid Password Valid choice	Login unsuccessful as an admin	3.1.1 Figure 8
13	Enter valid admin Username and valid Password	Registered admin	Valid Username Valid Password Invalid choice	Login unsuccessful as a restaurant owner	3.1.1 Figure 8

	Select the restaurant owner choice				
14	Enter valid admin Username and invalid Password  Select the restaurant owner choice	Registered admin	Valid Username Invalid Password Valid choice	Login unsuccessful as an admin	3.1.1 Figure 8
15	Enter invalid admin Username and valid Password  Select the restaurant owner choice	Registered admin	Invalid Username Valid Password Valid choice	Login unsuccessful as a restaurant owner choice	3.1.1 Figure 8
16	Enter Invalid admin Username and invalid Password  Select the restaurant owner choice	Registered admin	Invalid Username Invalid Password Valid choice	Login unsuccessful as a restaurant owner choice	3.1.1 Figure 8
17	Enter valid Username and valid Password Choice = NULL	Registered admin	Valid Username Valid Password NULL choice	Login unsuccessful (Select a choice)	3.1.1 Figure 8
18	Enter invalid Username and valid Password Choice = NULL	Registered admin	Invalid Username Valid Password NULL choice	Login unsuccessful (Select a choice)	3.1.1 Figure 8
19	Enter valid Username and invalid Password Choice = NULL	Registered admin	Valid Username Invalid Password NULL choice	Login unsuccessful (Select a choice)	3.1.1 Figure 8
20	Enter invalid Username and invalid Password Choice = NULL	Registered admin	Invalid Username Invalid Password NULL choice	Login unsuccessful (Select a choice)	3.1.1 Figure 8
21	Enter valid Username and valid Password Select the restaurant owner choice	Unregistered user	Valid Username Valid Password Valid choice	Login unsuccessful (Should Register)	3.1.1 Figure 8
22	Enter Invalid Username and valid Password	Unregistered user	Invalid Username Valid Password Valid choice	Login unsuccessful (Should Register)	3.1.1 Figure 8

	Select the restaurant owner choice				
23	Enter valid Username and Invalid Password Select the restaurant owner choice	Unregistered user	Valid Username Invalid Password Valid choice	Login unsuccessful (Should Register)	3.1.1 Figure 8
24	Enter valid Username and valid Password Select the restaurant owner choice	Unregistered user	Valid Username Valid Password Valid choice	Login unsuccessful (Should Register)	3.1.1 Figure 8
25	Enter valid Username and valid Password Select the admin choice	Unregistered user	Valid Username Valid Password Valid choice	Login unsuccessful (Should Register)	3.1.1 Figure 8

## DECISION TABLE

Test Case	Condition							Expected Output
	Valid Username for owner	Valid password for owner	Valid Username for admin	Valid password for admin	Registered Restaurant owner	Registered Admin	Unregistered User	
1	T	T	-	-	T	-	-	Login as a Restaurant owner
2	T	F	-	-	T	-	-	Error
3	F	T	-	-	T	-	-	Error
4	F	F	-	-	T	-	-	Error
5	T	T	-	-	-	T	-	Error
6	T	F	-	-	-	T	-	Error
7	F	T	-	-	-	T	-	Error
8	F	F	-	-	-	T	-	Error
9	-	-	T	T	-	T	-	Login as an Admin
10	-	-	T	F	-	T	-	Error

11	-	-	F	T	-	T	-	Error
12	-	-	F	F	-	T	-	Error
13	-	-	T	T	T	-	-	Error
14	-	-	T	F	T	-	-	Error
15	-	-	F	T	T	-	-	Error
16	-	-	F	F	T	-	-	Error
17	T	T	-	-	-	F	-	Error
18	T	F	-	-	-	F	-	Error
19	F	T	-	-	-	F	-	Error
20	F	F	-	-	-	F	-	Error
21	-	-	T	T	F	-	-	Error
22	-	-	T	F	F	-	-	Error
23	-	-	F	T	F	-	-	Error
24	-	-	F	F	F	-	-	Error
25	T	T	-	-	-	-	T	Error
26	T	F	-	-	-	-	T	Error
27	F	T	-	-	-	-	T	Error
28	F	F	-	-	-	-	T	Error
29	-	-	T	T	-	-	T	Error
30	-	-	T	F	-	-	T	Error
31	-	-	F	T	-	-	T	Error
32	-	-	F	F	-	-	T	Error



## PART 2

### TOOLS

#### 1. Selenium

##### DESCRIPTION

Selenium is a framework designed for automating the testing of web applications, offering broad compatibility with various browsers, operating systems, and programming languages. It excels in cross-browser testing for web applications (JMeter vs Selenium: What Is Preferred by Testers, n.d.).

##### FUNCTIONALITIES

Selenium, a powerful framework for automating web application testing, offers a range of functionalities through its various components, with Selenium WebDriver being the most prominent.

1. **Cross-Browser Testing:** Selenium allows testing web applications across multiple browsers such as Chrome, Firefox, Safari, and Internet Explorer.
2. **Automation of User Interactions:** Selenium can automate user interactions with web elements, such as clicking buttons, filling forms, and navigating through pages.
3. **Parallel Testing:** Selenium Grid allows parallel execution of tests on multiple machines and browsers simultaneously.

##### STRENGTHS

1. Ability to build custom frameworks.
2. Support for parallel testing through Selenium Grid.
3. Flexibility for code customization.
4. Support for Multiple Programming Languages: Java, Python, C#, Ruby, and JavaScript

## WEAKNESSES

1. Challenges in test maintenance with AJAX applications.
2. Outdated UI
3. Does not support testing of mobile applications.

Selenium is best suited for automation and cross browser testing of a website (JMeter vs Selenium: What Is Preferred by Testers, n.d.).

## 2. Apache JMeter

### DESCRIPTION

Apache JMeter, is a Java-based open-source application designed for conducting load tests and assessing the performance of software. Originally intended for web application testing, it has expanded its capabilities to cover various testing functions (JMeter vs Selenium: What Is Preferred by Testers, n.d.).

### FUNCTIONALITIES

JMeter is a tool that helps discover the maximum number of concurrent users a website can handle by simulating multiple user accesses concurrently. It performs load testing to model expected usage and stress testing to determine the maximum load capacity of a web server.

In load testing, various companies simultaneously send requests to a client, and JMeter measures the application's performance under heavy data load during the response.

1. Normal Load: The typical usage scenario where the application is subjected to a standard level of user activity.
2. Heavy Load: Testing under conditions of maximum expected usage, simulating a high volume of concurrent user accesses.
3. Target in the Test: The specific objectives and performance benchmarks that the test aims to achieve.

These considerations help in designing effective tests to evaluate how a web application performs under different scenarios, ensuring its reliability and stability under normal and peak usage.

## STRENGTHS

1. User-friendly UI.
2. Easy installation on different operating systems.
3. Integration with Jenkins, and support for various testing types like API, Database, and MQ testing.
4. It's also capable of handling a high number of transactions per second.

## WEAKNESSES

1. **Automation Challenges:** Automation can be challenging with JMeter.
2. **Complex Reports:** Output reports may be difficult to interpret without proper training.
3. **Limited JavaScript and AJAX Support:** JMeter lacks support for JavaScript and AJAX requests.
4. **Challenges with Dynamic Content:** Testing complex applications with dynamic content or those using JS to alter requests can be difficult.
5. **Limited Customization:** It might be challenging to extract data from specific sources or perform customizations.

Despite these limitations, JMeter remains a popular choice for performance testing due to its ease of use and broad functionality (JMeter vs Selenium: What Is Preferred by Testers, n.d.).

### 3. Junit

#### DESCRIPTION

JUnit is a widely-used open-source framework in the Java programming language for performing unit testing. Java developers rely on JUnit to write and execute automated tests, ensuring the integrity of their code with each new addition. The framework provides graphical representations, with graphs displaying green for successful tests and red for failures, allowing developers to create reliable and bug-free code.

#### FUNCTIONALITIES

Its key functionalities include test annotations (e.g., `@Test`), assertions for result validation, and test runners for automated execution. JUnit supports parameterized tests, exception testing, and the creation of test suites for organized execution. Rules and categories add flexibility and organization to tests. Overall, JUnit provides a robust set of features for developers to create reliable and efficient test cases for Java applications (K, 2022).

#### STRENGTHS

1. **Various Test Mechanisms:** It provides text-based command lines and graphical test mechanisms.
2. **Annotations:** JUnit offers annotations that simplify the implementation of test functions.
3. **Test Runner:** It includes a test runner to execute and manage test applications.
4. **Automated Testing:** JUnit facilitates automated testing, providing automatic feedback on code behavior.

#### WEAKNESSES

1. **Dependency Testing Limitations:** JUnit lacks comprehensive support for dependency testing compared to frameworks like TestNG.
2. **Less Suitable for Large Test cases:** It may not be the best choice for extensive test cases or higher-level testing.
3. **Limited Group Testing:** JUnit does not support group testing.
4. **HTML Reports:** JUnit does not inherently create HTML reports for test cases; external tools are needed for this purpose.

In summary, JUnit is a powerful tool for Java developers, offering a balance of advantages and limitations in the realm of unit testing (K, 2022).

## DISCUSSION

Area	Selenium	Apache JMeter	JUnit
<b>Primary Use</b>	Automated testing of web applications.	Load testing and performance assessment of software.	Unit testing in Java.
<b>Compatibility</b>	Broad compatibility with various browsers, OS, and languages.	Originally designed for web application testing.	Java-based, widely used for Java applications.
<b>Testing Types</b>	Web application testing, cross-browser testing.	Load testing, stress testing, performance testing.	Unit testing, integration testing.
<b>Key Component</b>	Selenium WebDriver.	JMeter application with various components.	JUnit framework with annotations.
<b>Parallel Testing</b>	Selenium Grid allows parallel testing.	Supports parallel execution of tests using JMeter Grid.	Limited support for parallel testing.
<b>User Interactions</b>	Automates user interactions with web elements.	Simulates multiple user accesses in load testing scenarios.	Focuses on unit testing, less on user interactions.
<b>Programming Languages</b>	Java, Python, C#, Ruby, JavaScript.	Primarily Java-based, supports plugins for additional languages.	Java

## USES OF SELENIUM

Selenium is a versatile framework primarily used for automating the testing of web applications. Its uses extend beyond just testing, and it finds applications in various scenarios. Here are some common uses of Selenium:

1. **Automated Testing:**

- **Functional Testing:** Selenium is widely used for functional testing of web applications to ensure that all features work as intended.
- **Regression Testing:** It helps in quickly detecting any unintended changes or regressions in the application after updates.

2. **Cross-Browser Testing:** Selenium supports testing across multiple browsers, ensuring that web applications behave consistently on different platforms.

3. **Performance Testing:** While not a dedicated performance testing tool, Selenium can be used to simulate user interactions and load on a web application.

4. **Browser Compatibility Testing:** Selenium is valuable for checking the compatibility of web applications across different browsers, ensuring a consistent user experience.

5. **User Interface (UI) Testing:** Selenium is commonly used for UI testing, verifying that the user interface elements and interactions are working correctly.

6. **Integration Testing:** Selenium is often used for integration testing, validating that different components of a web application work together seamlessly.

7. **Headless Browser Testing:** Selenium supports headless browser testing, allowing tests to run without a graphical user interface, which can be useful for faster test execution.

These diverse applications showcase the flexibility and utility of Selenium in various aspects of web development and quality assurance (Selenium Automation Testing - a Comprehensive Guide by HeadSpin, n.d.).

## BIBLIOGRAPHY

- *20+ Best Open-Source Automation Testing Tools in 2020.* (n.d.).  
Www.softwaretestinghelp.com. <https://www.softwaretestinghelp.com/open-source-testing-tools/>
- *Best Free Open-Source Automation Testing Tools & Frameworks.* (n.d.). Katalon.com.  
<https://katalon.com/resources-center/blog/open-source-testing-tools>
- *Function Point Analysis - Introduction and Fundamentals.* (2020, January 9). Fingent Technology. <https://www.fingent.com/blog/function-point-analysis-introduction-and-fundamentals/>
- *JMeter vs Selenium: What is preferred by Testers.* (n.d.). BrowserStack.  
<https://www.browserstack.com/guide/jmeter-vs-selenium>
- K, V. (2022, April 29). *What is Junit and How it works? An Overview and Its Use Cases.*  
DevOpsSchool.com. <https://www.devopsschool.com/blog/what-is-junit-and-how-it-works-an-overview-and-its-use-cases/>
- *Selenium Automation Testing - A Comprehensive Guide by HeadSpin.* (n.d.).  
Www.headspin.io. Retrieved October 20, 2023, from  
<https://www.headspin.io/blog/selenium-testing-a-complete-guide#:~:text=Selenium%20is%20a%20popular%20open>
- *Top 5 Open-Source Automation Testing Tools in 2023.* (2022, September 13). Preflight Blog  
No-Code Web App Testing. <https://preflight.com/blog/open-source-automation-testing-tools/>

## MARK ALLOCATION SHEET

After having discussed as group, we recommend the following mark allocation to each group member based on contribution or lack of it throughout the assignment.

**Group Name** – A2\_ S11199815\_ S11198967

Member ID	Percentage contribution of allocated task
S11199815	100%
S11198967	100%

Certification

ID	Member Name	Signature
S11199815	Savindya Fernando	<b>Savindya</b>
S11198967	Karan Parmar	<b>Karan</b>